www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

A PILOT MODEL FOR A PROOF OF CONCEPT HEALTHCARE FACILITY INFORMATION MANAGEMENT PROTOTYPE

SUMITTED: September 2012
REVISED: February 2012
PUBLISHED: March 2013 http://www.itcon.org/2013/5
EDITOR: Kirti Ruikar

Jason Lucas, Assistant Professor,
Department of Construction Science and Management, Clemson University;
jlucas2@clemson.edu

Tanyel Bulbul, Assistant Professor,
Department of Building Construction, Virginia Tech;
tanyel@vt.edu

Walid Thabet, Professor ,
Department of Building Construction, Virginia Tech;
thabet@vt.edu

SUMMARY: *Within the healthcare industry it is important that facility information is efficiently and effectively managed to aid in the successful operation and maintenance of the facility and provide a safe and well-maintained environment of care for patients and staff. In order to do this, a healthcare facility information management prototype was proposed. The goal of the prototype is to allow facility managers to more efficiently and effectively respond to facility related events within the healthcare environment. The prototype was designed by using case analysis methods to identify information needs and draw connections between clinical information and facility management operations. This information was organized into a product model which is used as an ontology to capture, store, and retrieve the information. A conceptual model is developed to demonstrate the potential use of the developed product model in aiding facility managers' responses to facility related events. The conceptual model uses developed Graphical User Interfaces (GUI) that are mapped to the product model to demonstrate the information interactions. Information from one of the analyzed case studies is used as an example to describe the conceptual model development. A walk-through of the model is included to demonstrate how the different GUIs would be used to respond to the example event. This paper discusses the system architecture, the goals of the conceptual model and prototype, and the conceptual model development and validation. Future research strategies for the proposed healthcare facility information management framework are also addressed.*

KEYWORDS: *Building Information Modelling, Conceptual Model, Healthcare, Facility Management, Information Exchange, Product Model.*

REFERENCE: *Jason Lucas, Tanyel Bulbul, Walid Thabet (2013) A pilot model for a proof of concept healthcare facility information management prototype, Journal of Information Technology in Construction (ITcon), Vol. 18, pg. 76-98, http://www.itcon.org/2013/5*

1. INTRODUCTION

Within healthcare environments there are many complex environmental systems that are important to the working of the clinical staff as well as patient treatment, recovery, and safety. The condition of these systems and the physical environment has been linked to the patient recovery, satisfaction, and overall well-being (Ulrich et.al., 2004; Harris et.al., 2002; Devlin and Arneill, 2003). It is the job of the facility management personnel to ensure that these systems are running properly and the physical environment is properly and safely maintained. Further complicating the task of managing the facility is limitations on personnel and resources and clinical operations are the revenue source for these healthcare environments. Effective facility management is needed in order to maintain the facility and these complex systems with minimal impact on the clinical operations that it houses.

1.1 Identified Problem and Motivation

Poor information management and missing data lead to wasted time and money throughout a facility's lifecycle (Deng, et.al., 2001) and can be traced to inadequate coordination of information caused by insufficient, inappropriate, inaccurate, inconsistent, or late information communication (Gallaher et.al, 2004, Eastman et.al, 2008). In order to support efficient and effective facility management, the facility information and other information types needed for conducting facility management activities need to be properly managed. However, the current information flow of needed information is disconnected. This disconnect occurs in two directions. Along the facility lifecycle from preconstruction/design, construction, and the operations/maintenance phases as well as between the different organizational groups that inhabit the facility during the operations/maintenance phase (clinical personnel/information and facility management personnel) (Lucas et.al, 2012).

In order to properly maintain the facility, facility management personnel need to not only know information about the systems that they maintain but they also need to know what is happening within that facility. Specifically, patient occupancy, identifying where patients are located in connection to where work is being performed to ensure proper containment, clinical occupancy, identifying what types of activities are taking place in each area of the facility, and clinical scheduling, specifying what areas are being used within the hospital at any given time. These types of information help to aid in planning maintenance activities as well as ensuring proper procedures are followed in emergency work situation. Studies have shown that repair, maintenance, and renovation activities can have an impact on patient safety and length of stay (Oren et.al, 2001; Loo et.al, 1996; Lutz et.al, 2003). Linking activities of the hospital to planning maintenance is not an easy task as schedules are constantly being updated and some maintenance activities cannot be planned. Typical solutions involve using email and phone communications between facility management personnel and clinical administration involved with scheduling procedures and bed allocation. These solutions are not quick and efficient enough, especially when there is an emergency maintenance situation. Access to real-time information would help create a more efficient response.

As shown in Lucas et.al. (2012) through case analysis, the location of an event in proximity to patients can influence the response needed for a given situation. Meaning, if a water leak appears in an operating room there are certain procedures that need to be followed that would not need to be followed if the leak appeared in the mechanical room in the basement. The analysis also demonstrated that the condition of the patient and the length of exposure to the situation can cause health problems, such as wound infections, that will extend the length of stay and also increase the risk of further complications in the patient's health and safety. If a patient is directly exposed to a condition, such as a leak, they are at greater risk than if the leak happened in a janitor's closet down the hall. When these types of situations occur that threaten the safety and care of patients, they are considered patient safety events. Patient safety events end up causing the United States healthcare industry $19.5 billion (USD) annually (Sheve et.al, 2010). They also increase the cost of patient care by 17% and extend patient stays by 22% longer than those that are not connected to a safety event (IOM, 1999).

1.2 Aim

The research, discussed in part in this paper, discusses a healthcare facility information management system that was proposed to offer an efficient and effective method for facility managers to capture, manage, retrieve, and use real-time clinical and facility information in response to emergency events (Lucas et.al, 2011). The ontology and product model proposed not only collects and stores information but aids in determining event response as well. As part of this research, documented cases from a 500 bed United States based hospital that serves over 27,000 annual admissions were used as a baseline study and for further analysis. Prior to this paper, the case study analyses were completed (Lucas et.al, 2012) and the data model developed (Lucas et.al, 2013). The purpose of this paper is to discuss how the user can interact with the data model through a developed conceptual model of a proof-of-concept prototype. How the data model and prototype can be further tested for effectiveness and impact is also discussed.

1.3 Current Research in Literature

Within healthcare environments, management of clinical and patient information through the use of information technologies and data models is prevalent with Healthcare Information Technologies (HIT) and Electronic Healthcare Records (EHR). Both HIT and EHR have been used with wide success to improve patient safety and clinical operations (Bates and Gawande, 2003).

Other clinical based initiatives include an ontology based system to aid in execution of clinical guidelines when caring for patients to help reduce errors and improve patient care by providing correct action plans to clinical personnel (Isern et.al, 2012), automating clinical pathways through semantic rules (Hu et.al, 2012), and a context-aware system used to monitor patient post-care conditions (Fenza et.al, 2012). These systems are developed to aid in clinical operations and help with decision support while determining the proper steps in response to a patient's health condition.

Within facility management and healthcare, the use of information systems consist mainly of efforts to use BIM as a coordination tool for conducting construction and renovation work (Khanzode et.al, 2008), for visualization and energy analysis (Sheth et.al, 2010), or for the use of healthcare building information systems like Johnson Controls' Metasys (http://www.johansoncontrols.com) or Notifier (http://www.notifier.com). Not specific to healthcare, the use of BIM and other information technologies for supporting facility management are most commonly involved with energy analysis (Woo et.al, 2010), capturing as-built data for facility management use (Tang et.al, 2010), and computation support for managing change (Akcamete et.al, 2000).

Ontology has been used within facility management, separate from healthcare, to semantically acquire, cleanse, transform, index, manage, and share knowledge through a formal representation (Lima et.al, 2005), verify consistency of computer interpretable material during construction (Staub-French and Nepal, 2007), document rational and knowledge during processes (Straub-French et.al, 2003), and for managing context-sensitive construction information (Wang et.al, 2010). Ontology within BIM development has also been used to define formal relationships between elements (Succar, 2009), help define design reasoning in space programming and sizing of rooms (Kim and Grobler, 2007), and for documenting pre-construction activities for reference through the lifecycle of the construction phase of the project (Lee et.al, 2008). An ontology-based framework and BIM had also been examined for real time data and decision processing during the operation of the building by looking at temperature readings, sensor readings, and alarm systems to collect data and perform immediate calculation and determine proper response (Tsai et.al, 2009).

Most similar to this last effort, the proposed framework will collect and manage real time clinical and facility management information and with support of historical and building data aid in determining the proper response. The difference is the proposed framework is dealing with facility management malfunctions within a clinical environment, not just how to manage environmental control systems. There is a gap in literature for managing facility information to support maintenance and repair response.

2. HEALTHCARE FACILITY INFORMATION MANAGEMENT FRAMEWORK

The healthcare facility information management framework targets the previously identified problem of fragmented information within healthcare facility management. This is done through the proposal of a method for efficient and effective information capture, management, query, and retrieval to support facility management personnel in responding to facility related events in a healthcare facility. The framework takes into account historical and real time facility and clinical information as well as healthcare regulations and standards in place within the United States.

In order to develop the framework, the following objectives were set:

1. Identify the critical information needed for proper event response.
2. Organize the critical information into a logical data model to define relationships of information and information exchange mechanisms.
3. Develop a conceptual model for the proof-of-concept prototype to test the developed data model.
4. Prototype development
5. Validation, evaluation, and testing
6. Deployment

To this point objective one and two are complete with main focus of this paper being on objective three with a discussion of the future steps including validation, evaluation, and testing of the prototype.

2.1 Framework Development Methods

In order to complete the set objectives different research and analysis methods were used. Objective #1 was completed through a rigorous case analysis process (Lucas et.al, 2012). Objective #2 involved the creation of a product model and information exchanges to help manage the information (Lucas et.al, 2013). Both objectives #1 and #2 have been completed and are summarized below.

In order to complete Objective #3, a system architecture defining the interactions of the user with the system was developed. In order to develop the Graphic User Interfaces (GUIs) needed for the prototype, Use Cases were developed. These use cases mapped the product model developed in Objective #2 to the GUIs. From here, the interfaces were designed for interaction with the user. Test cases were used to help validate the product model within the conceptual model to check that all necessary information was retrieved. Completion of Objective #3 is discussed in greater detail below.

2.1.1 Identifying Information Needs through Case Analysis

In order to identify the information that needs to be incorporated into the framework, case based analysis methods were used. Possible case topics were documented through meetings with the clinical and facility personnel of a 500 bed university hospital. To narrow the scope of initial development, cases involving mechanical systems that had a level of threat to patient safety were selected out for further analysis. Case narratives for the selected cases were developed to document step by step procedures that were followed in responding to the cases. These case narratives were reviewed for accuracy by the hospital personnel. Several case analysis methods were used to analyse the case narratives and determine the information needs for each response (Lucas et.al, 2012).

First, narrative was documented as a process model in Business Process Model Notation (BPMN). This allowed for the analysis of each individual task and decision that is made during the process as it occurred while linking it to the actor (system, individual, or work unit) responsible for that action. From here, Failure Mode and Effects Analysis (FMEA) and Fault Tree Analysis (FTA) were used to extract data and define relationships between facility data and clinical data. The FMEAs allowed for analysing the systems involved in the event for other failures and effects than those that were present in the original case. The FMEAs were used as a basis for the FTAs that help to determine possible route causes for each of the failures. Lastly, Unified Modelling Language (UML) is used to define Use Cases for analysing each step of the base case. The FMEAs were used to determine alternate flows from the original base case within the Use Cases. The Use Cases were then mapped to determine information needs that existed throughout the event response. The information was mapped back to its origin within the lifecycle to determine what types information needs to be captured and managed through the different lifecycle phases (ex. HVAC Air Handler manufacturer and model # from construction, etc.).

One case study that was analysed and used as an example in this paper is the “Malfunctioning HVAC in the Operating Room”. In this case there was a water incursion caused by a malfunctioning HVAC unit above the operating suite within a university hospital facility. There was water coming out of the coil connected to the chiller supply line. It was determined that a blockage was created due to oxidation on the inside of the coil. The water was first noticed by an operating room nurse who was getting supplies from the sterile storage room. The water was dripping from the ceiling and pooling on the floor and had also damaged the wall. The damaged wall was connected to an operating room. It was noticed later that the water had also soaked through the floor and had caused ceiling and wall damage to an emergency department patient bay on the floor below. All areas had to be contained and closed off. The nurse notified the building control center who paged the maintenance mechanic to the scene. When on the scene, the mechanic quickly tried to contain the water damage. From there, he had to locate the source of the problem and shut it down before repairs could be done. During the response, an event response team was organized to aid in the decision making process and make sure that the proper repairs were done. The affected services were cleared within 48 hours of the event reporting. When all repairs were done, it was estimated that there was $7 million (USD) in damages, repairs, lost supplies, and displaced revenue.

2.1.2 Product Model and Ontology Development

Once the case analyses were completed a list of vocabulary terms was compiled. This led to the creation of a product model (Fig. 1) (Lucas et.al, 2013). The product model is used to store the information. Information exchange mechanisms were documented as UML Sequence Diagrams to show the behaviour of the classes and information interactions between the classes of the product model. The design scope of the product model is to allow for facility management personnel to have adequate information in an efficient and effective way to support their response to facility related events.

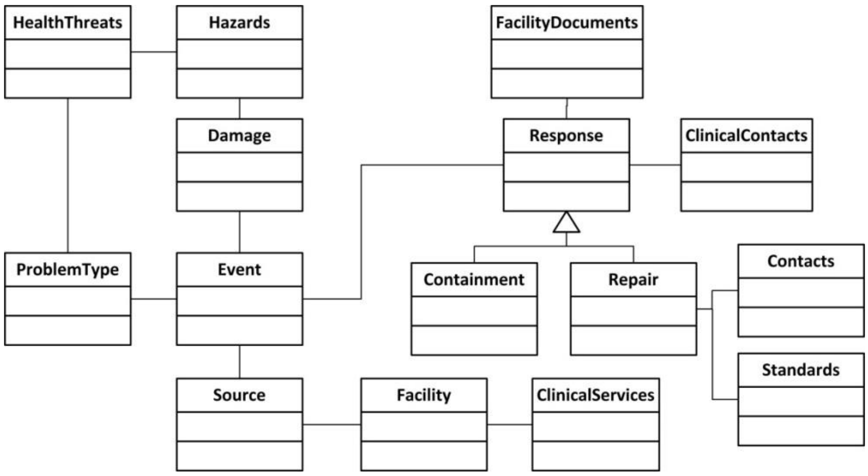

FIG. 1: Class Diagram of Product Model

The core of the product model is the `Event` class. From the information that is documented within the `Event` class the proper response can be determined and the source of the problem can be found. The main attributes for determining the response and source are the problem type, symptoms, and location of the reported event. Connected to the `Event` class is the `Source` class which has a relation to the `Facility` class. The `Source` class allows for using the location and symptoms to find components within the facility at the designated location that can cause the problem. The `Facility` class adapts the structuring of facility related information from the Construction Operations Building Information Exchange (COBie) (East, 2012). COBie is used in this research since it is a developing standard that many facility managers and other building professionals have some familiarity with as a method for documenting facility related information for use during the operation and maintenance of the facility. Also connected to the `Event` class is the `Response` class with sub-classes for `Containment` and `Repair`. The `Containment` stores information and operations for identifying different hazards and health threats and the risk levels for infection and damage. The `Repair` class helps to organize the actual repairs based on the damages. Information within the `FacilityDocuments` class pertains to protocols, work order systems, and similar facility related documents. Its relationship with the `Response` class helps to determine the proper response based on the situation. Associated with the `Event` class is also the `Damage` class. Different types of damages are stored in this class. Related to the `Damage` class is the `Hazard` class. Each `Damage` instance has relevant hazards that need to be accounted for. Connected to the `Hazard` class is the `HealthThreats` class. Each hazard can have relevant health threats. Each problem type can also have relevant health threats to it.

3. CONCEPTUAL MODEL FOR A HEALTHCARE FACILITY INFORMATION FRAMEWORK

In order to demonstrate how the product model can potentially be accessed by a user and be used to aid in repair response, a conceptual model of a proof-of-concept prototype was designed. This conceptual model includes designed GUIs that users would use to actively work with the product model to retrieve data and plan a well-informed response to a facility related event. The conceptual model demonstrates the usability of the product model as a mechanism for accesses relevant information during the facility management response to patient safety events.

The prototype conceptual model uses the case study "Malfunctioning HVAC in Operating Room" as an example to describe its implementation. As was the case in developing the product model and class interactions, scalability for inclusion of other systems that were not involved in the initial design use cases is kept in mind. It is expandable for inclusion of additional building systems when the framework is expanded in future work. A test case was used to validate the design of the concept model is flexible for situations beyond the cases used in the research.

3.1 System Architecture

Fig. 2 shows the overview of the system architecture for the prototype. The interactions between the different pieces of the prototype are designed within the conceptual model.

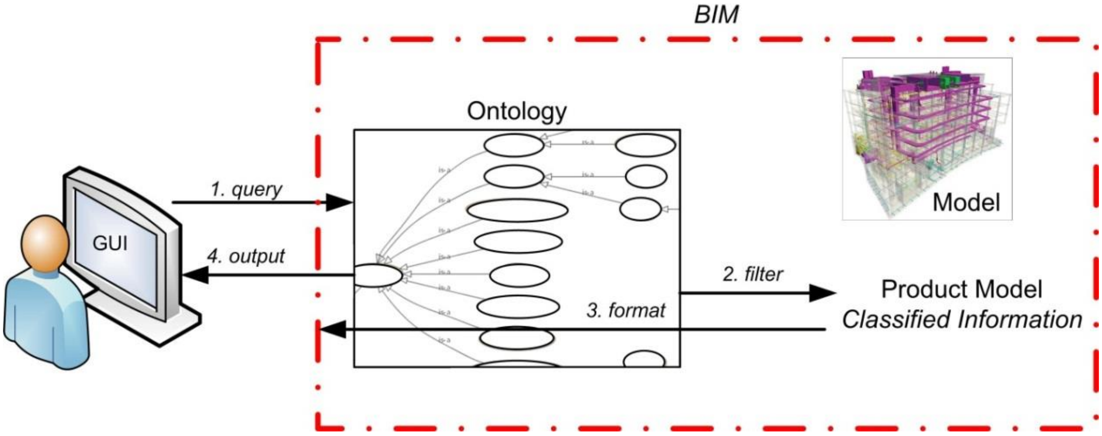

FIG. 2: System Architecture Overview

Within the system architecture, the GUI allows for the user to define basic information needs and the framework generates the information the user needs. The developed product model serves as a container for storing information. Connected to that container is a set of information exchange mechanisms that allow for filtering, querying, and accessing different information based on the user input. The information exchange mechanisms, acting as an ontology, define how classes interact with each other and are represented in the prototype as operations that run in the background coding. They also output requested information back to the user.

3.2 Implementation Criteria

Implementing the conceptual model for a product model serves two purposes, (1) to demonstrate how the product model can be used to access information and (2) help validate the information contained within the product model. In order to demonstrate how the product model can be used, GUIs had to be developed that allow the user query information and review data.

The intended user of the system is facility management personnel, especially a facility management mechanic. From the initial interviews and discussions with the healthcare facility management personnel it was determined that not many of them are familiar with BIM-based software and complex computer systems. This heavily influenced the design of the GUIs that are developed. The programs that they are currently familiar with using for other facility management activities are button and menu based, so similar methods to access the information were developed within the GUIs. This was important so the user may more easily learn how to use the system.

Another criterion for implementation is to make sure that the processes and sequence are within the regulation and familiar to what is done now. The goal is to make the process more efficient, however, with so many variables involved, the process that the system takes the user on, needs to ensure all variables are taken into consideration. The process cannot be reinvented because of the regulatory procedures involved that can only be determined by carefully examining specific variables.

Lastly, since the conceptual model will be used for the prototype implementation it is designed with the use of Java (http://www.java.com) in Eclipse (http://www.eclipse.org). This combination helped to develop the prototype interface that will allow the user to interact with the rest of the framework. The main scope of the developed prototype conceptual model remains within the scope of the currently developed product model with a demonstrated use to access mechanical systems related information during a patient safety event to support the users' response to the event.

Future implementation will involve connecting the prototype to an IFC compliant model within an open-source system such as openBIM (http://www.openbim.org) or BIMserver (http://www.BIMserver.org) that allows for integration of an IFC model and a Java environment. Not included with the current implementation but will be included as criteria in the future are methods to manage and maintain the framework. These issues are beyond the scope of the current implementation.

3.3 Conceptual Model Development

The conceptual model and prototype are designed around the intended use of supporting facility management response to safety events and intended user being facility management personnel. The development cycle for the conceptual model was as follows:

1. Use Case development: use cases were developed to detail the interactions between the GUI, developed product model, and other systems. The use cases helped to organize the operations performed by the system and ensure that the GUIs are being developed within the defined scope and intent.
2. GUI mapping: Using the use-cases as a basis, the seven main GUIs that are included within the conceptual model were mapped. This mapping identified the interactions between the GUIs and the product model. The interactions include background operations and information that is being input, exchanged, and output to the user through the GUI. The GUI mapping helps to organize the individual functions and define the programming that will be needed to obtain those functions.
3. Paper Prototyping: Paper prototyping was used as an efficient and effective method of sketching, organizing, and reviewing GUIs and their functions without the need for formal GUI development.
4. Conceptual Model: GUIs were developed within Eclipse. Functions were then mapped to the GUIs and diagrammed. Classes within the product model were connected to the GUIs to represent where the information would be retrieved from and how the information exchanges are handled.
5. Test-case validation: The test-case validation uses two additional case studies to test if the conceptual model design is flexible for use beyond the initial design case. Test-case information is mapped on the developed GUIs to check that all information needed in the additional cases could be handled by the designed conceptual model.

3.4 Use Case Development

The use cases were developed in the UML Format. UML is a visualization language that is used to model systems. It allows for specifying, visualizing, constructing, and documenting systems (Alhir, 1998). The Use Cases allow for documenting each individual step of the process. In the case of the design of the conceptual model, the different use cases document the actions of the GUI with rest of the developed framework. The use cases walk through the step by step interactions of the user with the GUI and the GUI with the product model. The information exchange mechanisms are represented as operations within the use cases. Each use case describes a different function of the prototype and became the base for the different GUIs. This allowed for organizing the GUIs so they incorporated the correct fields and features for the user to define specific aspects of the event and in return show the requested information through the GUI. The use-cases helped to ensure that the GUIs were not overcomplicated and only the necessary features to complete the task were included.

3.5 GUI Interaction Mapping

Once the use cases were developed there was an understanding of the types of interfaces that were needed and their functionality. In total, seven main GUIs were developed from the use cases. The GUIs were separated to show the different functions of the prototype. In order to organize their functions on paper and track the information exchanged, a GUI Interaction Map was completed. This map is organized as a table with four columns (Fig. 3 shows a partial table). The columns from left to right are *Work Order Documentation*, *Prototype GUI (User)*, *Background*, and *Product Model Store*. These columns represent where information is coming from, what operations are running, and how the GUI interacts with the product model and work order systems.

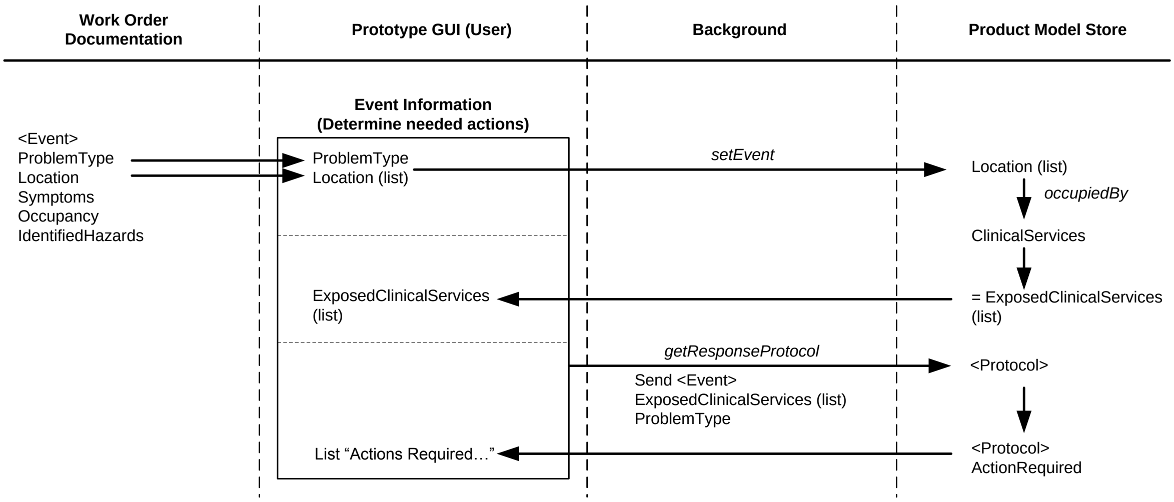

FIG. 3: Partial GUI Interaction Table

Under the *Work Order Documentation* column is information that is originally reported and found within the work order that is associated with the event. If the framework is connected to the building operation system that contains the work order information, this information can automatically be input to the first GUI and stored within product model when it is needed. The *Prototype GUI (User)* column contains the separate GUIs within an outlined box under a title for each. In this case, the “Event Information” GUI is included. The operations of the system that are running within the background are represented under the *Background* column. Lastly, the *Product Model Store* column represents the information exchange mechanism that input, store, format, sort, and retrieve information from the product model.

The labels within the table represent the information classes and attributes of the product model where the information is stored or retrieved from. The bracketed labels represent the classes while the labels listed under them are the separate attributes. For example `<Event>` is the event class within the product model and “ProblemType”, “Location”, “Symptoms”, “Occupancy”, and “IdentifiedHazards” are attributes of that class.

The arrows represent information moving between the different systems. Operations are listed in italic along the arrow. Information that is involved in an operation is listed as an input from the GUI (such as “ProblemType” for the *setEvent* operation). If the information does not come from the GUI and is from the product model it is listed under the operation as can be seen in the *getResponsePrototcol* operation. If an attribute can hold a list of items it is listed to the right or directly below the attribute in parenthesis as scene with “ExposedClinicalServices (list)”.

Within Fig. 3, from the *Work Order Documentation*, the “ProblemType” and “Location(list)” information, are the inputs from the `<Event>` class into the GUI. Using Case Study 1 as an example, this would be a water incursion event that occurred within the operating suite. The “Location” would consist of a list of spaces including the Operating Room, Sterile Supply, and Emergency Department Bays. This information is represented to the user within the GUI and used to start the event. The operation *setEvent* is run and within the background and product model the relationship of information is used to determine that

<ClinicalServices> that occupy the affected spaces can determine the list of "ExposedClinicalServices" that is in turn output back to the GUI. The same GUI is also used to determine the proper response through the *getResponseProtocol* operation.

Each of the GUI's interactions and information exchanges were mapped using this format. The combination of the GUI Interaction Maps and Use Cases allowed for an understanding of the type of programming that would be required within the prototype and helped to lay out the conceptual model.

3.6 Paper Prototype

The GUIs were sketched on paper to lay out how each of them would take user input and then show the user outputs from the framework. The overall prototype layout is set up with the user input on the left and the display window shown on the right (Fig. 4).

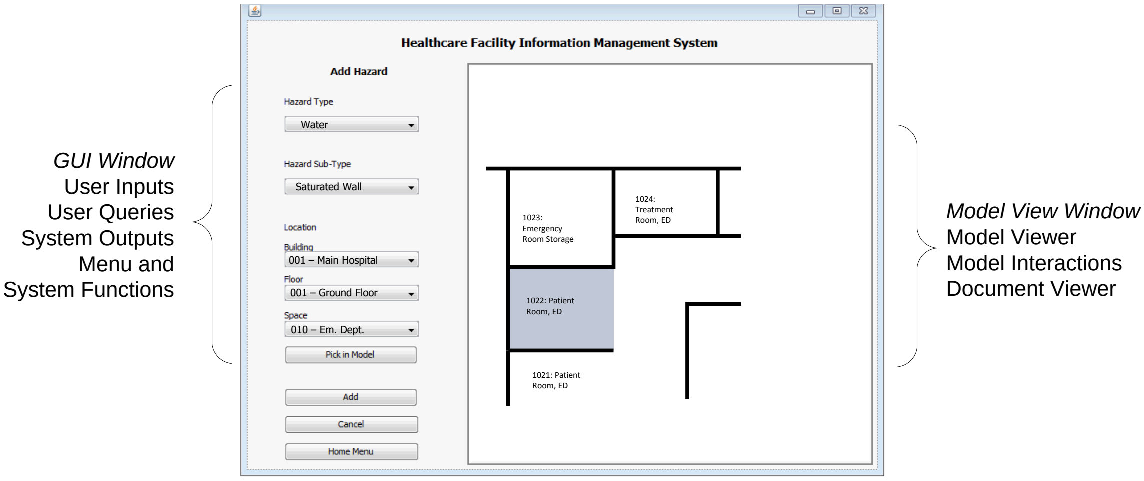

FIG. 4: Prototype GUI Layout

On the left side of the prototype is where the main inputs and outputs from the system are located. The right side of the prototype is a display window that allows for the user to interact with the system and information from the model is viewed. The GUI inputs and their functions for the left side of the prototype are summarized below:

1. <u>Event Information</u>: Classifies the type of event and determines the steps that are required based upon the problem type and location of the event.
2. <u>Hazard Mitigation</u>: Determine the mitigations that are needed for the associated hazards of the event that have been identified.
3. <u>Locate Source</u>: Based on the location and symptoms, the GUI and associated operations output possible sources that can cause the symptoms and are in the affected area. Linked to the possible sources is how to check and ensure each is working properly. The user can identify which of the sources is the problem and retrieve information on how to best shut down the affected system.
4. <u>Identify Risk and Damage Levels</u>: Based on the location and problem, the risk and damage levels can be determined using information that is available through the Emergency Operation Plan protocol. The GUI, using the needed information already input into the prototype automates the determining of the Risk and Damage Level. Also based on these variables a list of clinical and facility personnel can be listed that need to be involved in the response process. This GUI is linked to the database of contact information as well as knowing who is on duty under each contact category to allow for automatic paging of specified personnel.
5. <u>Damages</u>: This GUI takes the broken component that has caused the problem as well as the materials or parts that need to be replaced. It also allows the user to document the different damages that are noticed such as ceilings and walls that need replacing from water damage. The GUI lists the identified damages on the left of the prototype while allowing users to select the components from the model on the visualization side of the prototype. The damages listed help to identify hazards and health threats as well as what repairs are needed.
6. <u>Hazards and Health Threats</u>: Hazards that were identified during the initial documentation of the event as well as hazards noticed during the response process and those associated with the damages are input

into this GUI. The hazards are processed for associated actions that are required to make sure they have all been mitigated or are under consideration for repair. The hazards, along with the location of the event, also help to determine health threats associated with the event and what the proper actions are to deal with them.

7. Repairs: This GUI takes the list of affected comments and helps the user to determine what work will be completed by in-house personnel and what work requires a contractor. When the repairs are divided, the replacement parts or materials needed for the in-house repairs are listed with a status of if they are in stock or need to be ordered. If the parts need ordering, supplier information is available through the framework. For contractor repairs, a list of qualified contractors for the type of work is available with contact information. The last part of this GUI is listing the types of testing that is associated with each the repairs to make sure that things like moisture levels and air quality are within the allowable standards.

Additional GUIs may be needed to support situations involving other facility systems that are currently outside of the scope of the developed framework. When the functions and layout of each GUI was determined on paper, it allowed for review and editing before it was formally developed within the conceptual model.

4. CONCEPTUAL MODEL WALK-THROUGH

Once the GUIs and basic interactions were designed on paper, the conceptual model was developed. The conceptual model uses the designed GUIs and maps the inputs and outputs to the product model and ontology. The GUIs used in the conceptual model were developed using Java in Eclipse environment. These same interfaces can serve as the base for programming the prototype in future research. For the purpose of the conceptual model and demonstrating its potential use and functionality of the framework only the GUIs were needed at this time. The GUIs were then mapped to the product model and annotated as to how the information from the product model is exchanged and manipulated in determining the correct outputs back to the user.

The conceptual model is organized with a "Home Menu" on the main menu screen. As seen in Fig. 5, the "Home Menu" consists of 7 different buttons that allow the user to access the different GUIs that perform different tasks throughout the response process. When the user selects one of the buttons, they would be taken to that GUI. For instance, when the "Event Information" button is selected, the user is taken to the "Event Information GUI" as seen in Fig. 6. The right side of the prototype consists of a *Model View Window* that allows for interacting with the model and displaying information to the user.

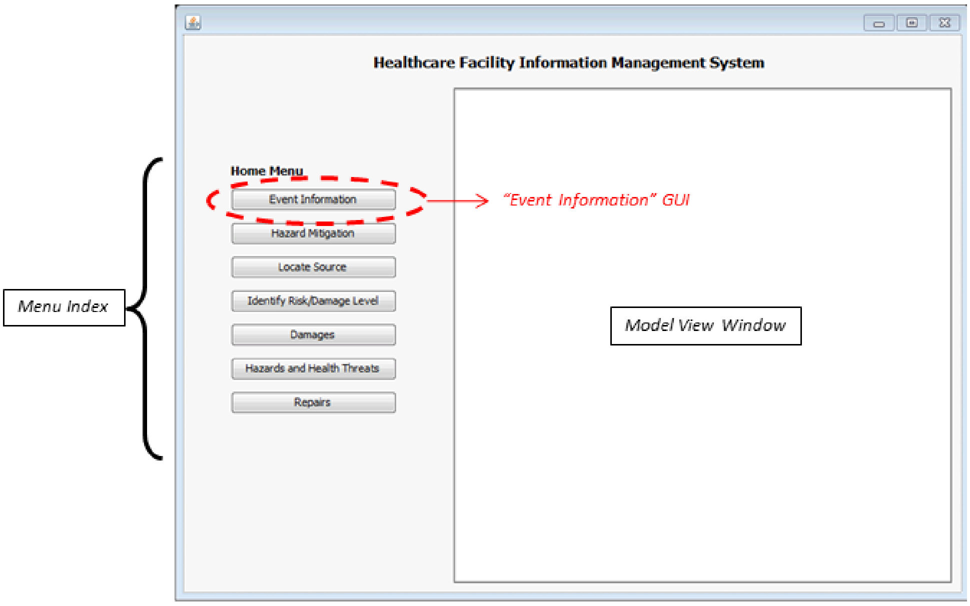

FIG. 5: Home Menu GUI

The "Event Information GUI" (Fig. 6) contains all of the information from the start of the event. Information for the "Problem Type", "Symptoms", and "Location(s)" is all available through the `Event` class within the product model. The Event class can be populated through the GUI or by information that is transferred by connecting the framework to the work order system that is used within the facility. If additional locations are discovered throughout the response process and event investigation additional locations can be added by the user through the "Add Location GUI" (Fig. 7). The "Exposed Clinical Services" are determined by the relationship between the ClinicalService and Facility class that classifies what clinical services occupy which space within the facility. The areas listed within the Location(s) attribute of the Event class would be equivalent to instances of the Space sub-class. The Locations that were identified in the example case are "2012 – Operation Room 3", "2013 – Sterile Supply 1", and "1022 – Patient Room, ED". These locations are instances of the Space sub-class under the Facility. Each of the Spaces is occupied by a ClinicalService, "General Surgery", "Central Supply", and "Emergency Department" respectively. These ClinicalServices become the list within the ExposedClinicalService attribute of the Event class.

The last area of the "Event Information" GUI is the "Immediate Actions Required". These are actions that the facility management personnel should take. They are based on the Protocol which is determined by using the ProblemType and ExposedClinicalServices to filter ActionsRequired which are the tasks to start the response process. In total when the "Water Incursion" problem type and listed exposed services are taken into account the response must include "Form Response Team", "Mitigate Hazards", "Diagnose Source", and "Shut-off Source".

The "Home Menu" button within the GUI will take the user back to the "Home Menu" GUI (Fig. 5).

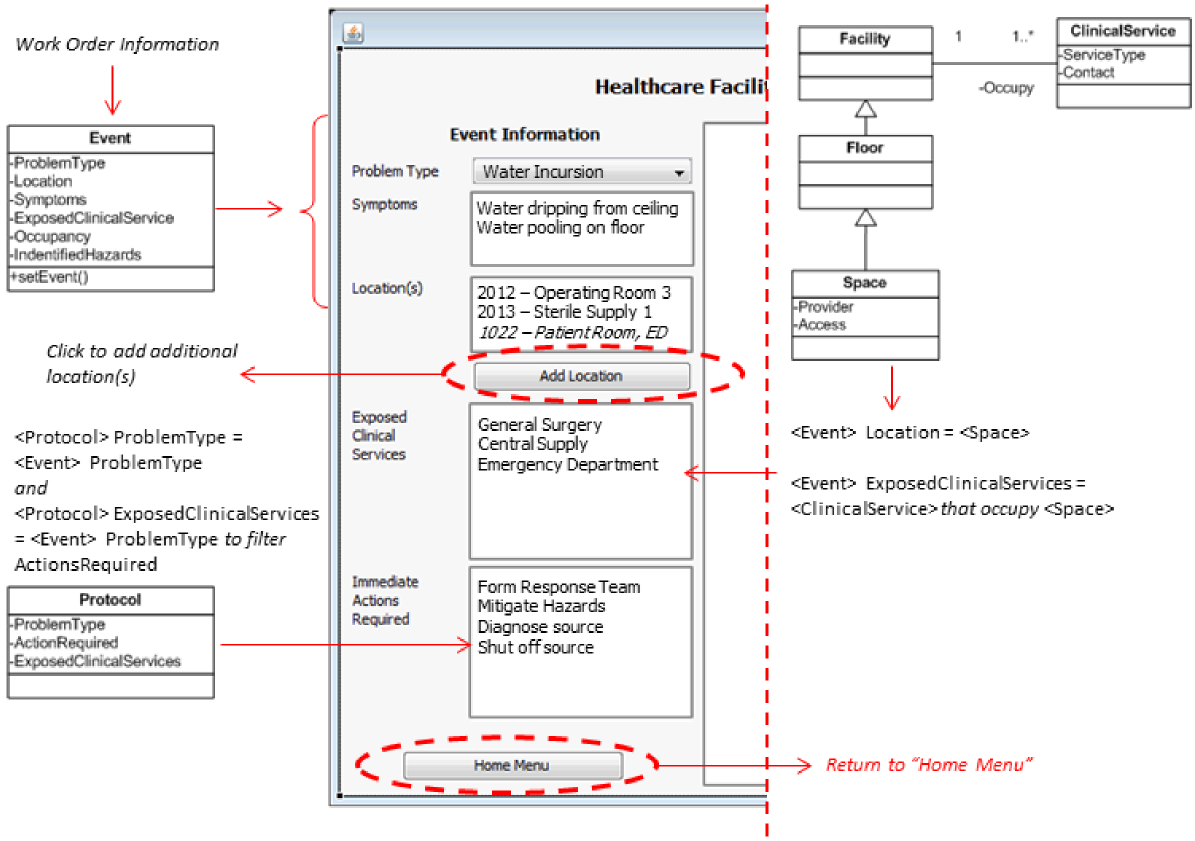

FIG. 6: Event Information GUI

From the "Event Information" GUI the user is able to "Add Location" (Fig. 7). This gives the user the ability to add locations that are involved in the event that may not have been immediately reported when the work order was created. Such as was the situation in the "Malfunctioning HVAC Unit in the OR" case where the initial report consisted of damage from water incursion within the operating room and sterile supply. Once the investigation was underway it was also determined that water had leaked through the floor and damaged the ceiling and walls of an emergency department patient bay. The GUI accounts for adding locations. Within the *Model View Window* of the GUI, the user can select the proper building, floor, and zone and then highlight the affected rooms within the model. This action is then recorded back on the "Event Information" GUI (Fig. 6) under the "Location(s)" and stored within the `Location` attribute of the `Event` class.

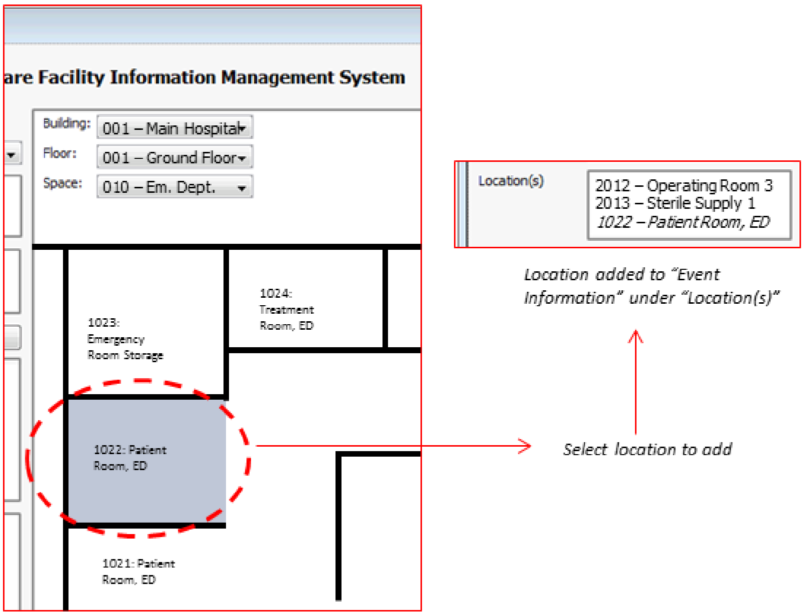

FIG. 7: Add Location GUI

The next GUI the user would use is the "Hazard Mitigations" GUI (Fig. 8). This GUI allows for a listing of the "Identified Hazards" which are written as *Location #: Category: Sub-Category*. For example the hazard "2013: Water: drip from ceiling" is an identified hazard in room 2013- Operating Room 3 and is a water incursion that has water dripping from ceiling. The identified hazards are stored within the `Event` class under the `IdentifiedHazards` attribute. Some hazards would be transferred from the Work Order System and identified in the initial event report while others would need to be added. The user can add hazards that are identified by clicking the "Add Hazard" button which would then activate the "Add Hazard" GUI (Fig. 9). In order to determine the "Actions Required" the `Type` of hazard is referenced to the `Hazard` class to filter out the `ActionRequired` to be listed on the GUI.

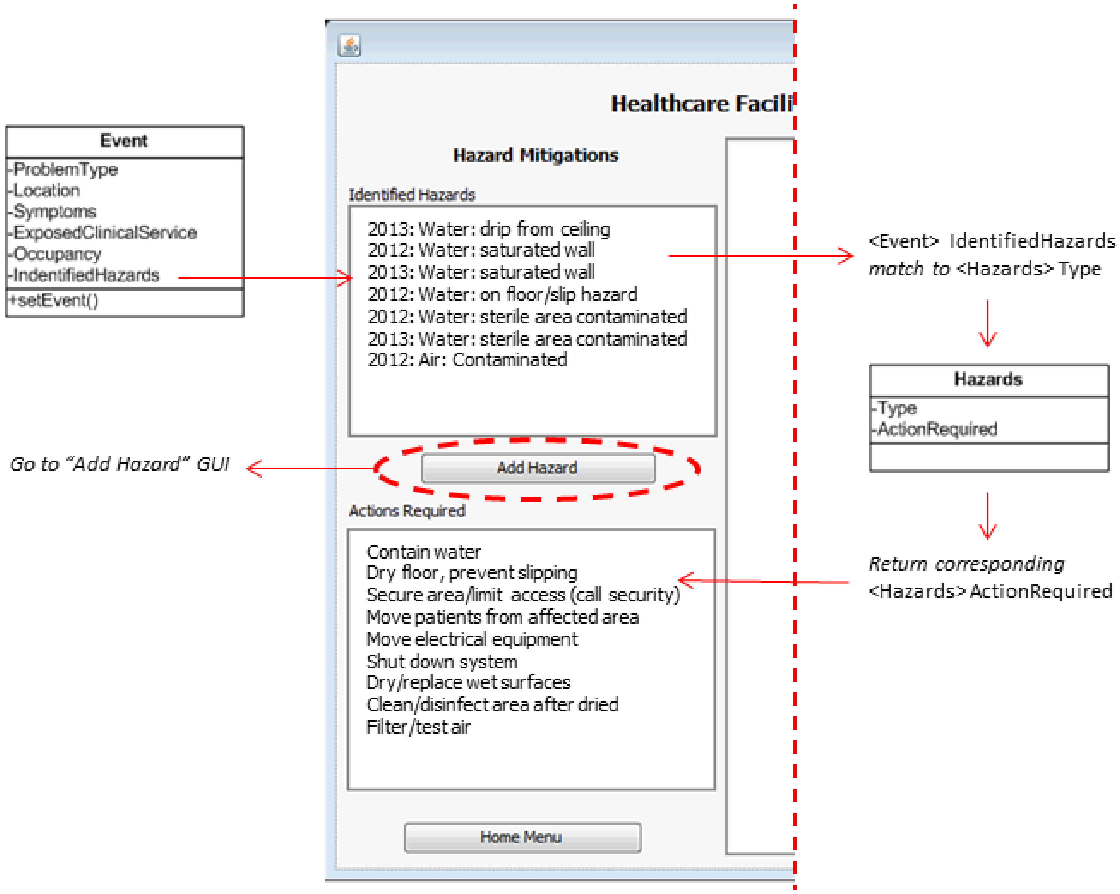

FIG. 8: Hazard Mitigation GUI

When an additional hazard is located, the user can add that hazard through the "Add Hazard" GUI (Fig. 9). This allows the user to identify the "Hazard Type" and "Hazard Sub-type" which would be instances of the `Hazards` class. It also requires the user to select the location. These locations available would be the ones that are already defined in the `Event` class.

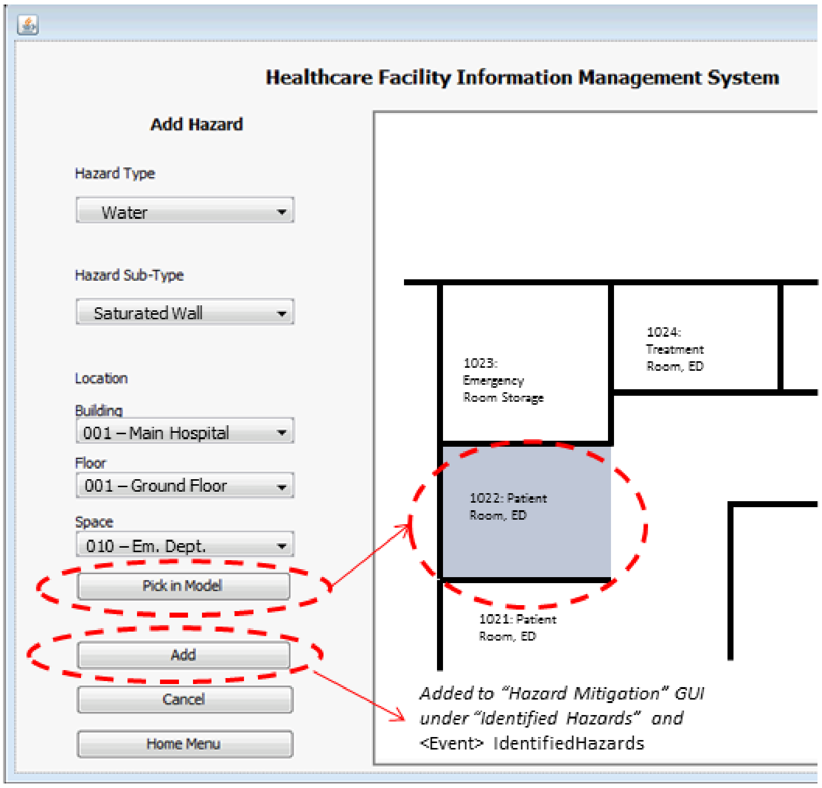

FIG. 9: Add Hazard GUI

Another GUI involved in the immediate response to the event is the "Locate Source" GUI (Fig. 10). This GUI takes into consideration the defined `Space` instances that are part of the `Location` of the `Event`. It then looks at instances of the `Component` class that are "locatedIn" the `Space` instances. This determines a list of components that are in the area of the problem. It then takes these instances of the `Component` class and looks for `PossibleFailures` that would match up to the `Symptoms` attribute of the `Event` class. In the example case, it looks at the "2012 – Sterile Supply 1" location to determine components of the facility that would cause "water from ceiling" as the `PossibleFailure` attribute. These components are then listed as `possibleSources` within the `Source` class and listed within the GUI under "Possible Sources". Facility personnel would need to look at each of the listed components and determine which one(s) of them is the actual problem. The GUI *Model View Window* is used to help the facility personnel view location information and other operation information for each of the components when they are highlighted. Once the user determines the problem, they select it from the list and click "Identified Source". This is then stored in the `brokenComponent` attribute of the `Source` class. `shutdownProcedures` for the `brokenComponent` are then viewable in the *Model View Window* based on the `Manuals` class relationship to the instance of the `Component` class that was identified.

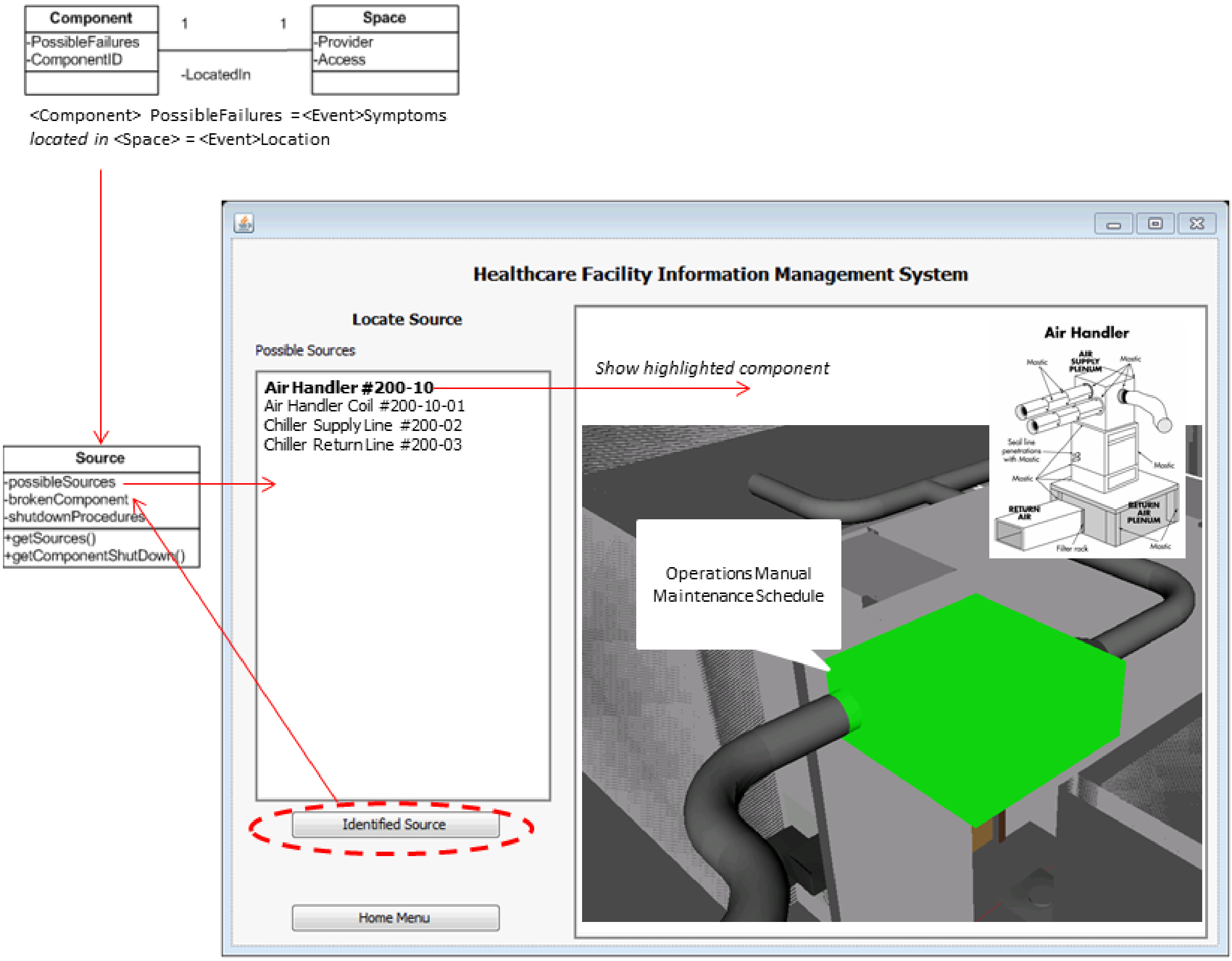

FIG. 10: Locate Source GUI

Another GUI is the "Identify Risk/Damage Level" GUI that is used to help determine the response team that needs to be formed to complete the mitigation and repairs. As a result of this GUI a Personnel Contacts list is developed of personnel who are needed to aid in the response. The "Event Infection Risk Level" and "Event Damage Level" are determined based on the `ProblemType`, `Symptoms`, and `ExposedClinicalServices` attributes of the `Event` class. These are then compared to the tables within the "EOP" (Emergency Operations Plan) instance of the `Protocol` class. As a result, by using the water incursion problem type in the operating suite and emergency department, the "Event Information Risk Level" is classified as "High" while the "Event Damage Level" based on the degree of symptoms for a water incursion problem is defined as "Level IV". The "Event Information Risk Level" is stored as `RiskLevel` and the "Event Damage Level" is stored as the `DamageLevel` within the `Containment` class. These are used along with the `ExposedClinicalServices` to filter instances of the `ClinicalContacts` class to determine the personnel who need to be included under "Personnel Contacts". Once this list is created the user can manually add a contact and send a page to all personnel notifying them of the location and problem type.

In addition to the broken component there may be other damages that exist and need repairs, such as some damaged drywall or ceiling tiles. The "Damages" GUI allows for listing all damages that were caused by the event that need to be repaired. The `brokenComponent` from the `Source` class is determined as the first damage. Additional damages can be added by the user by selecting the "Add Damages". All damages are then stored as `AffectedComponents` within the `Repair` class.

The "Add Damages" GUI (Similar to the "Add Hazard" GUI in Fig. 9) is used to add additional damages to list of "Noticed Damages". The user can define the location were the damages have occurred and then the type of damage based on the `Damage` class. The user can then select the component within the *Model View Window* to link the damage to the component. The damages are then added to the `AffectedComponents` within the `Repair` class.

Knowing the instances of the `Damage` class that are involved in the event will update the "Identified Hazards" lists and also determine the "Health Threats" in the "Hazards and Health Threats" GUI. Some instances of the `Damage` class are causes of instances of the `Hazards` class which causes the list of hazards, `IdentifiedHazards` in the `Event` class, to be updated. The `Hazards` can also cause health threats to patients and patient safety issues. This relationship between the `Hazards` and `HealthThreats` classes allows for determining the types of health threats that are involved in the event that administration need to be aware of and may take specific precautions to limit or mitigate. The health threats are also sometimes different depending on the clinical service within the area of the event. For that reason the `ClinicalServices` listed within the `ExposedClinicalService` of the `Event` class are also used to filter relevant health threats.

The last GUI developed within the concept model helps to organize the repairs that are needed. The "Repairs" GUI (Fig. 11) allows the user to separate the repairs between contractor and in-house repairs. The "Repairs" are listed based on the damages that were determined and stored within the `AffectedComponents` attribute of the `Repair` class. The user is able to select which repairs would be "C", contractor repairs, or "I", in-house repairs. From here the user can select to view either set of repairs by selecting the "In-house Repairs" button to activate the "In-House Repairs" GUI (Fig. 12) or the "Contractor Repairs" button to activate the "Contractor Repairs" GUI (Fig. 13).

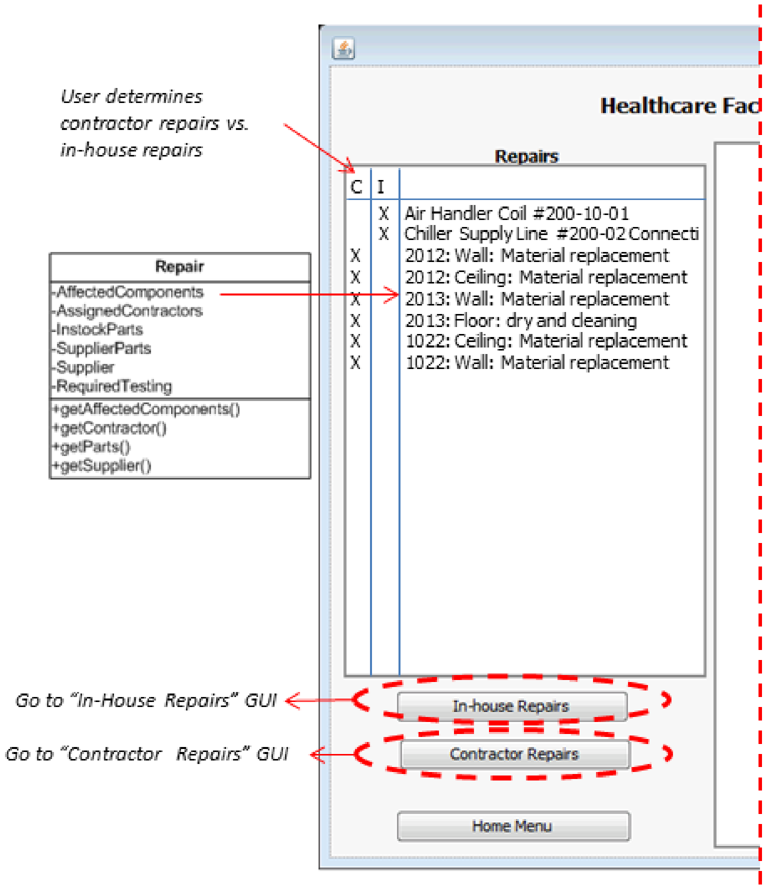

FIG. 11: Repairs GUI

The “In-house Repairs” GUI (Fig. 12) uses the *Model Viewer Window* and lists the repairs by part along the bottom of the screen. The parts are noted if they are in-stock along with a room number. If the part is not in stock then a supplier is listed. Clicking on the supplier gives the user the contact and order information or would allow them to find an alternate supplier. Clicking on the part would show the part and related information within the upper part of the *Model Viewer Window*. This information can then be used to make the repairs.

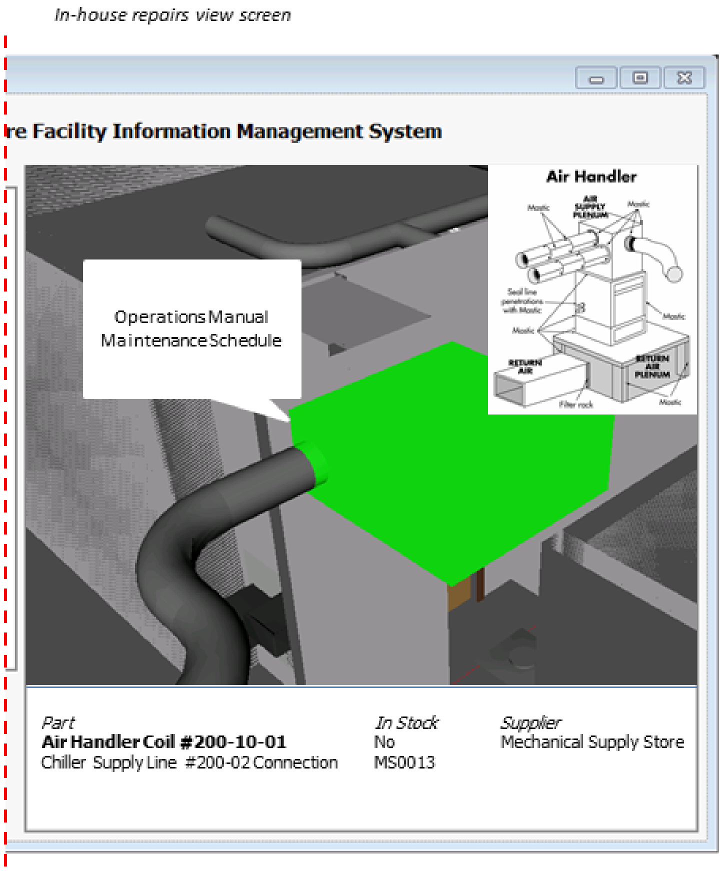

FIG. 12: In-House Repairs GUI

The “Contractor Repairs” GUI (Fig. 13) lists the different areas of work that need to be repaired. Along with each repair is a contractor that is able to do that type of work with contact information. The type of repairs and contractors are sortable. The GUI helps to organize the start of the repairs that need to be completed.

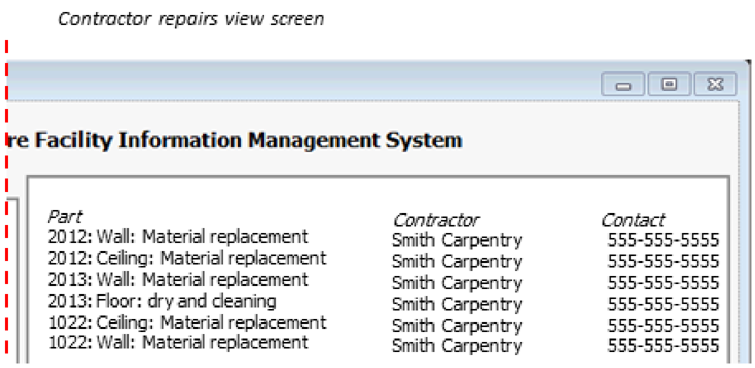

Fig. 13: Contractor Repairs GUI

5. CONCEPTUAL MODEL TEST CASE ANALYSIS

Once the conceptual model was designed and used to demonstrate the needs of the “Malfunctioning HVAC in Operating Room” case study, test cases were used to analyse the conceptual model for flexibility in incorporating other facility related patient safety events.

The first test case was the case study entitled “Chiller Pipe Burst/Air Conditioning Shutdown”. In this case, the outside temperature was consistently high for several days causing the HVAC system to work at its full capacity. There was a complaint to the facility managers that the temperature of a bank of patient rooms was too hot. A maintenance mechanic was sent to the scene to diagnose the situation and determine what the problem was. When he arrived in the unit, it was clear that even though the fan was working within the air handler, the air was not coming out cold. He then checked the chiller line and noticed that there was no pressure and no cold water coming through the system. Upon going into the basement the mechanic realized that there was a break in the chiller line and water was covering the basement floor. The entire system had to be shut down. Besides the damage to the basement due to the water and the need to remove parts of a wall to repair the break, the building was without air conditioning. With the exterior air being above 90 degrees, the interior quickly heated. The administration put into action a plan to transport all patients to other healthcare facilities in the area. Upon completion of the repairs to the main chiller pipeline, a terminal cleaning of the entire building and air quality testing was needed. The building was slowly cooled and dehumidified as to not cause any other damage to the HVAC system. 14 days went by before the facility was completely operational again.

The information from this case was overlaid into the developed prototype to check to make sure all needed information would be available. The “Event Information” holds the problem type of temperature with symptoms being too hot in the patient rooms. “Exposed Clinical Service” and “Immediate Actions Required” are determined as they were in the conceptual model walk-through. The “Hazard Mitigation” GUI serves the same function as the walk-through and allows for identifying hazards and mitigation methods needed.

The area where there is a slight difference is locating the source. Within the case study, the actual cause of the temperature raising is a chiller line burst in the basement from the main chiller supply line. Within the prototype, possible sources are determined by looking at components in the affected area that will cause the problem. It would be feasible to recognize that the chiller liner temperature and pressure is below what it should be in the area where the problem was originally detected, but the ultimate cause would need to be backtracked. During the case analysis phase of the project, when the systems were examined for causes of problem they were linked back to the root causes. These root causes would need to be connected to possible causes within the product model in order to locate the ultimate cause of the problem. The larger system of where the component is located would need to be examined. This system to component relationship is already defined within the product model. It would just need to be linked within the “Locate Source” window that is developed within the prototype conceptual model. This would need to be handled by adding an optional button. Instead of the “Identified Source” button that was in the initial conceptual model design, a “Search Root Cause” button was added (Fig. 14). This takes the selected possible source and gives information about the system that it is a part of, in this example one of the options would appear as “Main Chiller Line”. From here the model would work the same as previously described and allow the user to continue the process by identifying the infection risk and damage levels, documenting damages, identifying health threats, and organizing repairs.

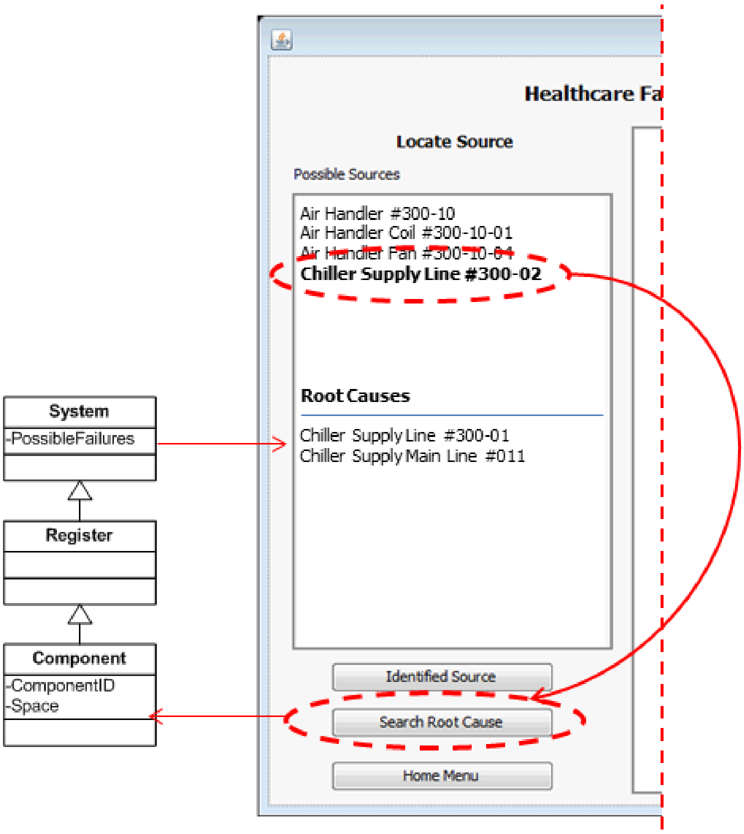

FIG. 14: Locate Source – Modified GUI

A second test case that was examined within the conceptual model is documented in Mohammadpour (et.al., 2012) and consists of a blockage in a main sewer line for a hospital. In this case, disposable anti-bacterial wipes and other materials that were not intended to be flushable were flushed into the system by environmental services personnel. The materials created a blockage in the main sewer line that exited the building. The situation was noticed by healthcare personnel when grey water started to accumulate along a floor drain on the ground floor of the hospital between central sterile and food services. Immediately food service and central sterile had to stop using their equipment that drained water into the line while facility management personnel worked at locating the source of the problem for fear that the sewer would back up into those locations. The blockage was found at the outside of the building between the lobby and the campus sewer main. Traditional drain snakes and augers that facility management personnel had on hand were not enough to remove the clog. Contractors were called in to help set up containment areas where the sewer line was located within the ground floor of the hospital. Pieces of a concrete wall needed to be removed to access the blockage and use a hydro-powered auger to clear the debris.

For this test case, the prototype revisions for the previous test case would be used. Within the response, the conceptual model allows for inclusion of problem type and location. This would give a list of protocol that needed to be followed. The hazards and mitigations would be accessible within the `Hazards` class. To locate the source, the "Search Root Causes" feature that was added would be used to access information for the system of the identified source at the problem location. The initial drain in that location and the pipes it was connected to was not the root cause. The root cause was a blockage near the sewer main. The damages, hazards and health threats, and repairs would all be organized as designed within the original conceptual model.

6. IMPLEMENTATION AND TESTING

The completed testing was used to say quantitatively if the framework, both the data-model and the prototype, does or does not allow the user to access the information needed during the response. The test cases helped to identify the areas that this was done well and the areas that needed improvement. These tests helped to ensure that the data model was not overly developed to try and include too much but also had correct and logical connection and relationships between information classes so information can be properly sorted and retrieved. The next step is to further develop the platform so it can be used for as a prototype in a pilot study to test its actual usefulness and impact. This will require additional implementation to expand the system beyond the identified mechanical system issues that were used for this preliminary research. As part of the implementation, the information and data will be validated by industry professionals. This would ensure that the correct information is retrieved for an inquiry. Once the implementation is complete, the testing can begin. There are several parts to this testing including a usability study, a usefulness study, and then an impact study.

The usability testing will examine how the intended users, the facility managers are able to interact with the system. This also involves how to represent the platform and will look at if a touch screen tablet, smart-phone, or laptop/computer station is the most appropriate. These usability studies will also look at how quickly the user can adapt to using the GUIs, ensure they are getting the information they are looking for, and make sure, based on subjective questionnaires, that the users are not frustrated with the process and they perceive it to be a benefit over traditional information retrieval practices. Preliminary meetings with facility personnel during the investigation stage of this research showed a positive reaction to the potential of such a system and these usability studies will help to better gauge its potential.

Once the usability study has been started and some feedback is returned, a usefulness study can be started. This usefulness study will look at the perception of facility managers toward the useful benefit of the platform. The study will take qualitative viewpoints of facility management personnel who would be using the system as well as their immediate supervisors and the facility administration and help gauge the acceptance of implementing the system. If the facility administration and staff do not accept the technology and believe it will be useful to make an impact, the chances are it will not be successful. Within this study, any issues that arise that would impact the acceptance of the system will be address either through education or further development/implementation changes.

Once the usability and usefulness studies with the prototype are complete, a limited deployment of the platform will be used for an impact study. The impact study will look at any changes in efficiency facility management personnel experienced by using the platform. This impact can be measured both qualitatively and quantitatively. Qualitatively through analysis of the system and if the facility management personnel believe the platform allowed for a more efficient and effective completion of required tasks and quantitatively by comparing response times to fix problems with and without the system. This quantitative analysis would be made possible only if the hospital has detailed records of response times and total time of repair of similar situations, most major healthcare facilities keep records of down time, the issue would be to locate a similar enough situation for the comparison. As part of the impact study would be to determine if the system has been fully adopted and what the time of adoption for the technology is. This would include making sure that the facility personnel are using the system as it was intended to be used and also measuring how long it takes them to become familiar enough with the system for its adoption to make an impact. These types of studies would help facility managers determine if this system is viable for wide spread deployment and let them determine if its use would benefit their facility.

7. LIMITATIONS

The platform discussed in this paper is still at a preliminary stage of development. Using the software lifecycle stages of Planning, Implementation, Testing, Documentation, Deployment, and Maintenance, it is only partly into the “testing” phase with the “planning” fully completed. The data model and prototype concept model have been tested through the use of use cases to ensure that the use can complete the needed steps and retrieve the proper information from the system during an event. It will, however, need to be implemented for use in a pilot study for additional testing before documentation, deployment, and maintenance.

At this time in its documentation, deployment, and maintenance features are very limited. It is intended that once the prototype is tested and before it is deployed, additional features will be built into the platform that allow for maintenance and editing of data. Documentation features will include the ability for a facility manager or the maintenance mechanic in charge to document the corrective measure within the data model. This will link the work order and the actions taken to the physical component in the BIM. This will also link any procedural information into the data model. It is intended that these type of responses can be tracked for future reference, so if a similar situation should arise in that or a different part of the hospital, it can help influence the steps taken to resolve the problem.

Finally, the system will need to be maintained and managed. With limitations on resources, it is not feasible to expect a facility to be able to hire someone to specifically manage this system. Ideally, the facility management and IT personnel that manage other computer systems within the organizations should be also to maintain this one. These individuals are in place at most large facilities as other computer based and automation programs are quite common. To help ensure the information stays accurate, general types of updates that will occur, such as regular maintenance or model renovations, will be examined to determine how they can be quickly and effectively integrated into the model. The intent is to make as easy as possible for a staff member with moderate computer knowledge and some training to maintain the system.

8. CONCLUSION

The proposed framework and conceptual model demonstrate how facility management personnel can efficiently and effectively access and manage information when responding to a facility related patient safety event. This framework links real time facility and clinical information to defragment current information technology systems. This is done to fill a gap that exists within the healthcare industry for facility managers who are responsible to responding to patient safety events in healthcare facilities. It does not just manage information around the physical components of the building but looks at procedural, managerial, and regulatory information to make sure that all areas are covered.

The success so far has been demonstrated with the conceptual model that exhibits a way of interacting with the data model. The test cases were used to validate the conceptual model to make sure that with known situations, the prototype would output the needed information and support the proper response procedures. Overall success and impact of the system will be gauged at many steps of future implementation and deployment with subjective analysis by potential users. The ultimate success would be defined when the impact study is complete. This would show any gain in efficiency of the systems use.

9. ACKNOWLEDGEMENTS

The background for this research is from a project funded under grant number 1 R02 HS19074-01 from the Agency for Healthcare Research and Quality (AHRQ), U.S. Department of Health and Human Services (HHS). The findings and conclusions in this document are those of the authors, who are responsible for its content, and do not necessarily represent the views of AHRQ. No statement in this manuscript should be construed as an official position of AHRQ or of the U.S. Department of Health and Human Services.

10. REFERENCES

Akcamete, A., Alkinci, B., Garrett, J.H., Jr. (2009), Motivation for computation support for updating building information models (BIMs). *Workshop on Computing in Civil Engineering 2009*, 523-532.

Alhir, S.S. (1998), *UML in a nutshell: a desktop quick reference*. O'Reilly and Associates, Inc. Sebastopol, California, USA.

Bates, D.W., Gawande, A.A. (2003). Improving safety with information technology, *New England journal of medicine*, Vol. 348, No. 25, 2526-2534.

Deng. Z.M., Li, H., Tam, C.M., Shen. Q.P., Love, P.E.D. (2001). An application of internet based project management system, *Automation in construction*, Vol. 10, No. 2001, 239-246.

Devlin, A.S., Arneill, A.B. (2003). Health care environments and patient outcomes: a review of the literature, *Environment and behaviour*, Vol. 35, No. 2003, 665-694.

East, E.W. (2012). Construction operations building information exchange (COBie), *Whole Building Design Guide, National Institute of Building Sciences, Construction* <<www.wbdg.org/resources/cobie.php>> retrieved on March 16, 2012.

Eastman, C.M., Teicholz, P., Sacks, R., Liston, K. (2008). *BIM handbook: a guide to building information modelling for owners, managers, designers, engineers, and contractors*, John Wiley and Sons, Inc., Hoboken, New Jersey, USA.

Fenza, G., Furno., D., Loia, V. (2012). Hybrid approach for context-aware service discovery in healthcare domain, *Journal of Computer and System Sciences*, Vol. 78, No. 2012, 1232-1247.

Gallaher, M.P., O'Connor, A.C., Dettbarn, J.L., Jr., Gilday, L.T. (2004). Cost analysis of inadequate interoperability in the U.S. capital facilities industry, *NIST GCR 04-867*, U.S. Department of Commerce Technology Administration, National Institute of Standards and Technology (NIST).

Harris, P., McBride, G., Ross, C., Curtis, L. (2002). A place to heal: environmental sources of satisfaction among hospital patients, *Journal of applied social psychology*, Vol. 32, No. 6, 1276-1299.

Hu, S., Li, J.S., Zhou, T.S., Yu, H.Y, Suzuki, M., Araki, K. (2012). Ontology-Based Clinical Pathways with Semantic Rules, *Journal of Medical Systems,* Vol. 36, No. 2012, 2203-2212.

Institute of Medicine (IOM), (1999), *To err is human: building a safer health system*. National Academy of Sciences, Washington, DC.

Isern, D., Sanchez, D., Moreno, A. (2012). Ontology-driven execution of clinical guidelines. *Computer Methods and Programs in Biomedicine*, Vol. 107, No. 2012, 122-139.

Khanzode, A., Fischer, M., Reed, D. (2008). Benefits and lessons learned of implementing building virtual design and construction (VDC) technologies for coordination of mechanical, electrical, and plumbing (MEP) systems on a large healthcare project. *ITCon*, Vol. 13, No. 2008, 324-342.

Kim, H., Grobler, F. (2007), Building Ontology to Support Reasoning in Early Design, Computing in Civil Engineering 2007, *Proceedings of the 2007 ASCE International Workshop on Computing in Civil Engineering*, 151-158.

Lee, J.S., Min, K.M., Lee, Y.S., Kim, J.H., Kim, J.J.(2008), Building Ontology to Implement The BIM (Building Information Modeling) Focused on Pre-Design Stage, *The 25th International Symposium on Automation and Robotics in Construction*. June 26-29, 2008: 350-354.

Loo, V.G., Bertrand, C., Dixon, C., Vitye, D., DeSalis, B., McLean, A.P., et.al. (1996). Control of construction-associated nosocomial aspergillosis in an antiquated haematology unit, *Infection control and hospital epidemiology*, Vol. 17, No. 6, 360-364.

Lucas, J., Bulbul, T., Thabet, W. (2011). A lifecycle framework for using BIM in healthcare facility management, *Computer knowledge building: CIB W78 W102 2011 proceedings,* 26-28 October, Sophia Antipolis, France.

Lucas, J., Bulbul, T., Thabet W. (2013). An object-oriented model to support healthcare facility information management, A*utomation in construction,* Vol. 31, No. 2013, 281-291.

Lucas, J., Bulbul, T., Thabet, W., Anumba, C. (2012). Case analysis to identify information links between facility management and healthcare delivery information in a hospital setting, *Journal of Architectural Engineering* (In press, accepted for publication September 2012).

Lima, C., El-Diraby, T., Stephens, J. (2005), Ontology-based Optimization of Knowledge Management in E-Construction. *ITcon*, Vol. 10, No. 2005, 305-327.

Lutz, B., Jin, J., Rinaldi, M.G., Wickes, B.L., Huycke, M.M. (2003). Outbreak of invasive *Aspergillus* infection in surgical patients, associated with a contaminated air-handling system, *Clinical infectious diseases,* Vol. 37, No. 6, 786-793.

Oren, I., Haddad, N., Finkelstein, R., Rowe, J.M. (2001). Interaction of processes and phases in project scheduling using BIM for a/e/c/fm integration, *Construction research congress 2010: innovation for reshaping construction practices*, 939-948.

Sheth, A.A., Price, A.D.F., Glass, J., (2010). BIM and refurbishment of existing healthcare facilities, *Proceedings of the 26th annual ARCOM conference, 6-8 September*, [ed. Egbu, C.], ARCOM, Leeds, United Kingdom, 1497-1506.

Shreve, J., Van Den Bos, J., Gray, T., Halford M., Rustagi, K., Ziemkiewicz, E., (2010). The economic measure of medical errors: sponsored by society of actuaries' health section, Milliman, Inc. (June).

Staub-French, S., Fischer, M., Kunz, J., Paulson, B. (2003), An Ontology for Relating Features with Activities to Calculate Costs. *Journal of Computing in Civil Engineering*,Vol. 17, No. 2003, 243-254.8

Staub-French, S., Nepal, M.P. (2007), Reasoning about component similarity in building product models from the construction perspective, *Automation in Construction*, Vol. 17, No. 2007, 11-21.

Succar, B. (2009), Building information modeling framework: A research and delivery foundation for industry stakeholders, *Automation in Construction*, Vol. 18, No. 3, 357-375.

Tang, P., Huber, D., Akinci, B., Lipman, R., Lytle, A. (2010). Automation reconstruction of as-built building information models from laser-scanned point clouds: a review of related techniques, *Automation in Construction*, Vol. 19, No. 2010, 829-843.

Tsai, W.T., Lee, W.H., Wiezel, A., Sun, X., Li, W. (2009), Ontology-based Service Composition Framework for Syndicating Building Intelligence, *2009 IEEE Conference on Commerce and Enterprise Computing*; 445-452.

Ulrich, T., Quan, X., Zimring, C., Joseph, A., Choudhary, R. (2004). The role of the physical environment in the hospital of the 21st century: a once-in-a-lifetime opportunity, *Report to the Center for Health Design for Designing the 21st Century Hospital Project*, September 2004.

Wang, H.H., Boukamp, F., Elghamrawy, T. (2010), An Ontology-based Approach to Context Representation and Reasoning for Managing Context-sensitive Construction Information, *Journal of Computing in Civil Engineering*, in press accepted October 8, 2010.

Woo, J., Wilsman, J., Kang, D. (2010). Use of as-built building information modelling, *Construction research congress 2010*, 538-548.